

BASES DE DATOS ORIENTADAS A COLUMNAS: UN ANÁLISIS COMPARATIVO FRENTE A LOS MODELOS DE BASES DE DATOS RELACIONAL, ORIENTADO A OBJETOS Y OBJETO RELACIONAL

Oriented Database Columns: A Comparative Analysis From Models Relational Database, Object Oriented and Object Relational

RESUMEN

Este artículo pretende identificar como las necesidades que deben satisfacer las soluciones de Bases de datos han cambiado en el tiempo y como con ello han surgido nuevos niveles de expresión para el modelado de datos, por tanto, es necesario entender las características de los modelos de bases de datos para así aplicarlos en dominios coherentes y eficientes.

PALABRAS CLAVES: Bases de datos, modelo, relacional, objeto, columna, SGBD, comparación, análisis, atributo, tipo.

ABSTRACT

This article aims to identify the needs that must be met Databases solutions have changed over time and thus have emerged as new levels of expression for data modeling, therefore, is necessary to understand the characteristics of the base models data in order to implement coherent and effective domains.

KEYWORDS: Database, model, relational, object, column, DBMS, comparison, analysis, attribute, type.

MARIA DEL PILAR AZCÁRATE TORO

Estudiante
Ingeniería de Sistemas y
Computación
Universidad Tecnológica de Pereira
pilar1601@hotmail.com

ALVARO CARDENAS OROZCO

Estudiante
Ingeniería de Sistemas y
Computación
Universidad Tecnológica de Pereira
alvarocardenas_0519@hotmail.com

1. INTRODUCCIÓN

En 1963 durante un simposio celebrado en california-Estados Unidos, se presentó por primera vez el concepto de base de datos, las aplicaciones ganaban día a día complejidad y era evidente la necesidad de dividir las aplicaciones de software en módulos especializados, durante este simposio se expuso entonces, que era posible separar los datos de las aplicaciones. Esta idea surge dado que la estructura de los datos no cambia en el tiempo significativamente, en cambio los lenguajes y plataformas tecnológicas si lo hacen, por tanto, podemos tratar el almacenado de datos y el que hacer de los mismos, como problemas aparte que posteriormente habremos de integrar, se definió entonces, una base de datos como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde entonces, se han presentado gran variedad de modelos que soporten plataformas tecnológicas para el almacenamiento de datos, y esto ha generado una competencia que ha traído consigo la evolución constante, tanto de los modelos teóricos, como de su implementación tecnológica.

Actualmente la evolución del internet, ha traído consigo nuevos retos desde la perspectiva del almacenamiento de los datos y de igual manera, las nuevas necesidades que aparecen en esta sociedad digital, es por esto, que nuestra investigación busca mostrar que diferencias trae consigo

la nueva tecnología de base de datos llamada Base de Datos Orientada a Columnas con sus predecesoras, y para ello, se habrá de expresar con rigor los modelos de bases de datos, desde su soporte teórico hasta su implementación tecnológica, caracterizarlos, y a partir de esto, realizar el análisis comparativo.

2. CARACTERÍSTICAS DE LOS MODELOS DE LAS BASES DE DATOS PROPUESTOS

2.1. Características del modelo de base de datos relacional

Codd [1] estableció en 1985 doce principios, de los cuales al menos seis deben satisfacerse para que una base de datos (BD) pueda considerarse totalmente relacional. Estos fueron precedidos de una regla general global, llamada "Regla Cero". Estos principios pueden resumirse de la siguiente forma:

Regla 0: Gestión de una BDR. Un sistema de gestión de base de datos relacional (SGBDR), debe ser capaz de manejar la BD exclusivamente con sus capacidades relacionales.

Regla 1: Representación de la información. Toda la información en una BDR se representa explícitamente a nivel lógico y de una manera única, por medio de valores en tablas.

Regla 2: Acceso garantizado. Todos y cada uno de los datos elementales en una BDR, deben ser accesibles lógicamente mediante el recurso a una combinación de: nombre de tabla, valor de clave primaria y nombre de columna.

Regla 3: Representación sistemática de la información que falta. Los valores nulos deben ser soportados por un sistema de gestión de BD (SGBD) completamente relacional para representar, de modo sistemático, la información desconocida o inaplicable.

Regla 4: Catálogo dinámico. La descripción de la BD se representa, a nivel lógico, en la misma forma que los datos ordinarios, de modo que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarlo.

Regla 5: Su lenguaje global de datos. Debe existir, al menos, un lenguaje cuyas sentencias sean expresadas por medio de una sintaxis bien definida, como cadena de caracteres, y capaz de soportar definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorizaciones y manejo de transacciones.

Regla 6: Actualización de vistas. Todas las vistas teóricamente actualizables deberán ser también actualizables por el sistema.

Regla 7: Inserciones, actualizaciones y eliminaciones de alto nivel. La capacidad para manejar, como un solo operando, la relación base o una relación derivada se aplica no sólo a las recuperaciones de datos, sino también, a sus inserciones, actualizaciones y eliminaciones.

Regla 8: Independencia física de los datos. Los programas de aplicaciones y las actividades terminales permanecerán lógicamente inalterados siempre que se realicen cambios en las representaciones de almacenamiento o en los métodos de acceso.

Regla 9: Independencia lógica de los datos. Cuando se efectúe en las tablas cualquier tipo de cambio que preserve la información, los programas de aplicación permanecerán intactos.

Regla 10: Independencia de la integridad. Las reglas de integridad de una BD particular deben ser definibles por medio del sublenguaje de datos relacional y almacenadas en el catálogo, no en los programas de aplicaciones.

Regla 11: Independencia de la distribución. Un sistema relacional debe tener un sublenguaje de datos que pueda soportar bases de datos distribuidas (BDD) sin alterar los programas de aplicaciones o actividades finales.

Regla 12: Regla de la no inversión. Si un sistema relacional tiene un lenguaje de bajo nivel, éste no puede

ser utilizado para pasar por alto las reglas de integridad y las restricciones expresadas por medio del lenguaje relacional de más alto nivel.

2.2 Características Del Modelo De Bases De Datos Orientado A Objetos

En 1989 Malcom P. Atkinson hizo el Manifiesto de los sistemas de base de datos orientados a objetos el cual propuso trece características obligatorias para un SGBDOO y cuatro opcionales. Las trece características obligatorias estaban basadas en dos criterios: debía tratarse de un sistema orientado a objetos y un SGBD [2].

Características obligatorias de orientación a objetos:

1. Deben soportarse objetos complejos
2. Deben soportarse mecanismos de identidad de los objetos
3. Debe soportarse la encapsulación
4. Deben soportarse los tipos o clases
5. Los tipos o clases deben ser capaces de heredar de sus ancestros
6. Debe soportarse el enlace dinámico
7. El DML debe ser computacionalmente complejo
8. El conjunto de todos los tipos de datos debe ser ampliable

Características obligatorias de SGBD:

9. Debe proporcionarse persistencia a los datos
10. El SGBD debe ser capaz de gestionar bases de datos de muy gran tamaño
11. El SGBD debe soportar a usuarios concurrentes
12. El SGBD debe ser capaz de recuperarse de fallos hardware y software
13. El SGBD debe proporcionar una forma simple de consultar los datos.

Características opcionales:

1. Herencia múltiple
2. Comprobación de tipos e inferencia de tipos
3. Sistema de base de datos distribuido
4. Soporte de versiones

2.3 Características Del Modelo De Bases De Datos Objeto Relacional

- Facilita el desarrollo de los pasos heurísticos que normalmente se hacen para diseñar una base de datos, ya que se pueden complementar fácilmente con el uso de UML, y las abstracciones y relaciones complejas para el desarrollo de un modelo lógico y físico más adecuado [3].
- Permite que los desarrolladores ahorren tiempo pues no tienen que escribir el código necesario que

comunique las tablas con los objetos pues este ya viene integrado en el SGBDR [3].

- Permite adaptarse a las necesidades de cada desarrollador porque es posible crear nuevo tipo de datos dependiendo del contexto y restricciones que se necesiten modelar [3].
- Reduce la cantidad de código escrito por lo menos un 20% aproximadamente respecto a las bases de datos relacionales [3].
- Si es necesario realizar transacciones de una forma ágil y consistente se puede aprovechar el soporte RDBMS contenido dentro del ORDBMS [3].
- Se soportan tipos complejos como: registros, conjuntos, referencias, listas, pilas, colas y arreglos [4].
- Se pueden crear funciones que tengan un código en algún lenguaje de programación como por ejemplo: SQL, Java, C, entre otros [4].
- Existe una mayor capacidad expresiva para los conceptos y asociaciones [4].
- Se pueden crear operadores lo que genera una mayor capacidad consultiva [4].
- Se soporta el encadenamiento dinámico y herencia en los tipos tupla o registro [4].
- Se pueden compartir varias bibliotecas de clases ya existentes, esto es lo que conocemos como reusabilidad [4].
- Posibilidad de incluir el chequeo de las reglas de integridad referencial a través de los triggers [4].
- Soporte adicional para seguridad y activación de la versión cliente-servidor [4].

2.3.1 Extensiones en las capacidades de los SGBDOR vs. SGBDR:

Carmen Costilla [5] propone las siguientes extensiones:

- Nuevos Tipos de Datos que permiten gestionar aplicaciones más complejas con una gran riqueza de dominios (imagen, voz, sueldo, etc.)
- Nuevas operaciones que permiten gestionar el comportamiento de los Tipos de Datos
- Mayor capacidad expresiva para los conceptos y asociaciones complejos Reusabilidad, propio de la Orientación a Objetos. Se pueden compartir diversas bibliotecas de clases ya existentes
- Integración de lenguajes: relacional, de objetos, XML, etc. En un solo lenguaje
- Nuevas Consultas con mayor capacidad consultiva (consultas anidadas, recursivas, almacenadas, pre-fabricadas), etc.

2.4 Características del Modelo de Bases De Datos Orientado a Columnas

- Es un modelo tecnológico orientado al almacenamiento en la base de datos en columnas en lugar de filas.
- En su mayoría están orientados a la optimización de la lectura de los registros que componen el almacén de datos.
- También se puede emplear el modelo para la optimización de la escritura de registros en un almacén de datos.
- Incluye memoria principal durante los procesos de consulta.
- Incluyen una implementación no tradicional de transacciones las cuales soportan una alta disponibilidad de aislamiento instantáneo para las transacciones de solo-lectura.
- En una arquitectura orientada a columnas basta con leer los valores de columnas necesarios para el procesamiento de una consulta determinada, y puede evitar poner valores de atributos irrelevantes.
- Para almacenar en columnas es necesario abordar una de las siguientes dos opciones:
 - Codificar los campos.
 - Usar variables comprimidas


3. ANALISIS COMPARATIVO ENTRE LOS MODELOS DE BASES DE DATOS PROPUESTOS

	1	2	3	4
Soporte a valores nulo	SI	SI	SI	SI
Actualización de vistas	SI	SI	SI	SI
Soportan objetos complejos		SI	SI	
Soportan mecanismos de identidad de objetos		SI		
Soportan encapsulación		SI		
Soportan clases		SI	SI	
Soporta herencia		SI	SI	
Soportan enlaces dinámicos	SI	SI	SI	SI
DML complejo	SI	SI	SI	SI
Conjunto de tipos modificable		SI	SI	
Gestión de bases de datos de gran tamaño	SI	SI	SI	SI
Soporta usuarios concurrentes	SI	SI	SI	SI
Recuperación de fallos	SI	SI	SI	SI

Convierte objetos complejos en tablas persistentes con el modelo relacional			SI	SI
Permite crear operadores		SI	SI	
Lenguajes de definición de datos unificados con lenguajes de programación		SI		
Consultas de mayor capacidad (anidadas, recursivas, almacenadas, etc...)		SI	SI	
Incluye memoria principal los procesos de consulta				SI
Codifica o comprime los campos para almacenarlos				SI
El almacenamiento está orientado a columnas				SI

- 1- MBDR
- 2- MBDOO
- 3- MBDOR
- 4- MDBOC

SI, significa que el modelo soporta el criterio.

 Significa que el modelo no soporta el criterio.

Partiendo de los modelos teóricos que soportan estos modelos de bases de datos, encontramos:

- El modelo de bases de datos Relacional esta soportado por el modelo relacional.
- El modelo de bases de datos orientado a objetos esta soportado por el modelo de objeto complejo.
- Los modelos de bases de datos objeto relacional y orientado a columnas son implementaciones tecnológicas que no tienen un soporte desde los modelos teóricos.

Comparando las diversas formas para la representación de la información, pudimos definir:

- El modelo de bases de datos relacional representa la información lógicamente por medio de valores en tablas
- El modelo de bases de datos orientado a objetos representa la información con objetos complejos.
- El modelo de bases de datos objeto relacional representa su información por medio de valores en tablas con valores complejos.

- El modelo de bases de datos orientado a columnas representa la información por medio de valores en tablas con valores complejos.

Comparando sus lenguajes de consulta identificamos:

- El modelo de bases de datos relacional soporta SQL2.
- El modelo de bases de datos orientado a objetos soporta ODMG-93 y SQL3.
- El modelo de bases de datos objeto relacional soporta SQL3.
- El modelo de bases de datos orientado a columnas soporta SQL2 y SQL3.

Observando el soporte teórico para la manipulación de los datos identificamos:

- El modelo de bases de datos relacional soporta algebra y calculo relacional.
- El modelo de bases de datos orientado a objetos soporta Algebra y cálculo de objeto complejo.
- El modelo de bases de datos objeto relacional soporta Algebra y calculo relacional extendido.
- El modelo de bases de datos orientado a columnas soporta Algebra y calculo relacional extendido.

Una diferencia fundamental entre los modelos de bases de datos y su implementación tecnológica es que mientras el modelo de bases de datos orientado a columnas esta optimizado para la lectura, los demás están optimizados para la escritura.

Desde la perspectiva de la facilidad para desarrollo de bases de datos identificamos:

- El modelo de bases de datos relacional se vale del esquema entidad-relación para modelar con suma facilidad nuestra base de datos.
- El modelo de bases de datos orientado a objetos permite modelar con naturalidad debido a su concepción de los objetos, y es particularmente eficiente cuando el sistema está pensado integralmente en objetos.
- El modelo de bases de datos objeto relacional ahorra durante el desarrollo toda la codificación necesaria para la conversión de objetos a tablas.
- El modelo de bases de datos orientado a columnas no tiene un modelo teórico que unifique la implementación tecnológica del mismo, por lo cual no existe un estándar que soporte la implementación específica de este modelo.

4. CONCLUSIONES

1. Desde la creación de los modelos de bases de datos, estos han evolucionado y a su vez han aparecido

otros, lo que ha generado el debate continuo sobre cuál modelo es el mejor, sin embargo en la actualidad se ha perdido interés en este debate dado que es claro que cada modelo tiene tanto sus ventajas como sus desventajas en determinados contextos, lo que ha causado que coexistan dado que:

- El modelo de bases de datos relacional es vigente hoy en día y si bien, a pesar de ser el más antiguo de todos, hoy por hoy es el modelo más utilizado por su facilidad de diseño y debido a su gran eficiencia en el área de las bases de datos transaccionales, las cuales cubren la mayor parte de las necesidades de desarrollo actuales.
 - El modelo de bases de datos orientado a objetos es vigente hoy en día, dado que cualquier transacción que involucre objetos complejos tendrá mayor eficiencia en un modelo orientado a objetos, ya que almacena directamente los objetos de la aplicación.
 - El modelo de bases de datos objeto relacional es vigente hoy en día, evidentemente debido a que al ser un modelo mixto que extiende el modelo de bases de datos relacional para soportar los objetos nos da una flexibilidad amplia para construir aplicaciones transaccionales que soporten objetos.
2. Las bases de datos orientadas a columnas son un concepto particular de la arquitectura y el almacenamiento físico de los datos no un modelo teórico propiamente. En otras palabras, no hay un modelo teórico único que soporte el funcionamiento de estas bases de datos y tampoco un estándar mínimo para su implementación.
- Las bases de datos orientadas a columnas son vigentes hoy en día dado que, han optimizado los accesos a disco durante los procesos de lectura y escritura lo cual genera una mayor eficiencia.
 - Como su nombre lo indica, guardan los datos en columnas en lugar de filas. Con este cambio ganamos mucha velocidad en lecturas, ya que si se requiere consultar un número reducido de columnas, es muy rápido hacerlo pero no es eficiente para realizar escrituras. Por ello este tipo de soluciones es usado en aplicaciones con un índice bajo de escrituras pero muchas lecturas.
3. Generalmente las aplicaciones del modelo de bases de datos orientado a columnas van orientadas a la optimización de consultas, por tanto se usan en entornos de poca escritura.

4. Podemos identificar los diferentes dominios de los modelos y plataformas tecnológicas para bases de datos, que permiten explotar su mayor potencial de la siguiente forma:

- El modelo de bases de datos relacional predomina en el campo de las bases de datos transaccionales.
- El modelo de bases de datos orientado a objetos ha demostrado su utilidad en campos como el CASE, CAD, sistemas multimediales y SIG, además de las diversas plataformas de diseño estructural e investigativas y el control en línea de robots.
- El modelo de bases de datos objeto relacional ha demostrado ser válido en contextos multimediales y SIG, pero además es bueno en sistemas transaccionales cuando se quiere minimizar la transformación de objeto a tabla.
- El modelo de bases de datos orientado a columnas muestra su potencialidad en contextos como el modelo de gestión de toda la organización, basado en la orientación al cliente (CRM), sistemas electrónicos de biblioteca de catálogos, bases de datos documentales y otros sistemas ad-hoc de investigación y típicamente es usado en data warehouse y sistemas de Business Intelligence (OLAP, etc.), donde además resultan ideales para calcular datos agregados.

5. Una diferencia fundamental entre los modelos de bases de datos y su implementación tecnológica es que mientras el modelo de bases de datos orientado a columnas está optimizado para la lectura, los demás están optimizados para la escritura

5. BIBLIOGRAFÍA

- [1] PONS Capote, Olga; MARÍN Ruiz, Nicolás; MEDINA Rodríguez, Juan Miguel; ACID Carrillo, Silvia; VILA Miranda, Ma Amparo. Introducción a las Bases de Datos. Thomson Editores Spain, 2005. p. 287.
- [2] Malcom P. Atkinson, Francois Banchon, Davis J Dewitt, Klaus R dittrich, David Maier, Stanley B zdonik – the object-oriented database system manifesto (1989).
- [3] GÓMEZ Gallego, Juan Pablo. Modelo Objeto relacional – ORDBMS. p. 7. Disponible en <<http://es.scribd.com/doc/270513/Bases-de-datos-Objeto-relacional>>.
- [4] ALBERCA Manzaneque, Alejandro; DÍAZ Tendero, Jesús. Modelos Avanzados de Bases de Datos. Funcionalidad 1. Bases de datos orientadas a Objetos y

Bases de datos objeto-Relacionales. p. 15. Disponible en
<<http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r23897.PDF>>.

[5] COSTILLA, Carmen. Apuntes Complementarios al Programa Docente, Tema V: El Lenguaje SQL de BDR, Estándar SQL: 99. Modelo de Datos Objeto-Relacional. 2005. p. 33. Disponible en
<<http://sinbad.dit.upm.es/docencia/grado/curso0506/Tran-sp%20Est%C3%A1ndar%20SQL-99-%20MD%20Objeto-Relacional.pdf>>.